

110793

31765

P. 8

## LESSONS LEARNED

### SUPPORTING ONBOARD SOLID-STATE RECORDERS

**Jeff Shi, Tony Mao, Tim Clotworthy**RMS Technologies, Inc.  
Code 520.9**Gerald Grebowsky**Data Systems Technology Division  
Code 521.1NASA, Goddard Space Flight Center  
Greenbelt, MD 20771

#### ABSTRACT

*With the advance of semiconductor technology, Solid-State Recorders (SSR) have matured and been accepted as primary onboard data storage devices. Their high reliability, simpler interface and control, and high flexibility have made the SSRs a superb choice in today's spacecraft design. While there are many benefits, the use of SSRs may also add significant complexity to ground data systems. For instance, real-time and playback data may be interleaved into the same data stream, making data sequencing and time ordering difficult. Stored data may be played back out of time order, increasing processing load significantly. Data may also be played back after being sorted by Virtual Channels in the SSR, potentially creating bursts in packet rates that exceed the real-time processing capabilities of the ground systems.*

*This paper presents a summary of lessons learned through the efforts in supporting a number of NASA's missions that employ SSRs. It describes various problems encountered through the design process, and their potential impact on ground system performance, resources, and cost. Recommended approaches to minimizing the impact are demonstrated by examples. The discussion leads to the conclusion that the use of SSRs demands an even higher level of cooperation between spacecraft and ground system designers in order to build the most cost-effective end-to-end system.*

#### 1. INTRODUCTION

With the advance of semiconductor technology, Solid-State Recorders (SSR) have matured and been accepted as primary mass data storage devices onboard spacecraft. Its high reliability, simpler interface and control, and high flexibility have made the SSR a superb choice for today's spacecraft designs. The use of SSRs also brings benefits to ground data processing systems. One of the most obvious advantages is the elimination of reversed data playback associated with the use of tape recorders. On the other hand, the flexibility of SSRs may also add complexity to ground data systems. For instance, real-time and playback data may be interleaved into the same data stream, making data sequencing and time ordering difficult. Stored data may be played back out of time order, increasing processing load significantly. Data may also be played back after being sorted in the SSR, potentially creating bursts in packet rates that exceed the capabilities of the ground systems.

The Data Systems Technology Division (DSTD) at Goddard Space Flight Center (GSFC) has provided support for a number of missions flying SSRs, including the Solar Anomalous and

Magnetospheric Particle Explorer (SAMPEX), Fast Auroral Snapshot Explorer (FAST), Submillimeter Wave Astronomy Satellite (SWAS), and Advanced Composition Explorer (ACE). In addition, planned application of SSRs in other missions such as the Earth Observing System (EOS-AM) have been reviewed.

The DSTD has developed a new-generation Packet Processing System (PPS) to perform science data processing for the FAST mission [1]. Scheduled for launch in August 1994, FAST is the second mission of the GSFC Small Explorer (SMEX) program, and one of the first missions based on Consultative Committee for Space Data Systems (CCSDS) recommended packet telemetry data formats [2]. Through the PPS development, extensive efforts have been made to support unique data scenarios generated with the flexibility of the SSR.

This paper presents a summary of lessons learned directly from FAST Packet Processing System development, and indirectly from reviewing other missions. Section 2 describes general characteristics of the onboard SSRs. Section 3 presents a detailed discussion of impacts, and recommendations. Section 4 summarizes the discussion with conclusions.

## **2. SSR CHARACTERISTICS**

Until recently, tape recorders have been the primary means to store data onboard a spacecraft. These recorders were used to store data when not in contact with ground stations, or communication satellites. Tape recorders were stream-oriented devices implying that data can only be accessed sequentially. To preserve power and increase the lifetime of tape recorders, data was typically played back without rewinding, generating a bit-reversed data stream during a playback pass. In addition, the recording and playback scenarios did not assure erasure of old data from the tapes. These scenarios often resulted in a number of old data fragments at either end of a data stream.

In contrast, the SSR is built upon Random Access Memory (RAM) technology and provides the capability for accurate and fast search, which is crucial for downloading data. Any set of stored data can usually be randomly accessed by selectable addressing in the memory. Accordingly, it offers the capability for more flexibility in data recording and playback. The obvious benefit to ground processing from the introduction of the SSR is the elimination of the tape recorder bit-reversed playback.

However, this same flexibility can lead to new challenges for ground data processing. This is especially true if the use of the SSR flexibility effects the sequential order, grouping, or block structure of recommended data formats. These problems are discussed in detail in Section 3.

## **3. IMPACT DEFINITION AND RECOMMENDATIONS**

One of the primary function required by ground data processing is the removal of artifacts introduced by the telemetry path to reproduce data that has the same form as data coming directly from the sensor. This implies that the data products should present data in its original time order with minimum errors and distortions. In support of this functionality, ground data processing incorporates the capability to detect and account for discrepancies in the expected order or structure of the received data. The potential impacts of the SSR application on ground data processing considered here fall into three categories:

1. Time order. The manipulation of time order, as described in the first three subsections, can increase system resource requirements and cost.

2. Data rate. When supporting the CCSDS packetized data format, the ground system's data processing capability or capacity is not simply defined by the input clock rate alone, but also by a packet rate. An example in Section 3.4 shows how under a constant clock rate, packet rates can push capabilities of ground processing systems to their limits.
3. Errors and gaps. The last two subsections are dedicated to discussions of errors and gaps induced during SSR operations, and the impact on the ground processing quality and accounting tasks.

### 3.1 MANIPULATION OF TIME ORDER

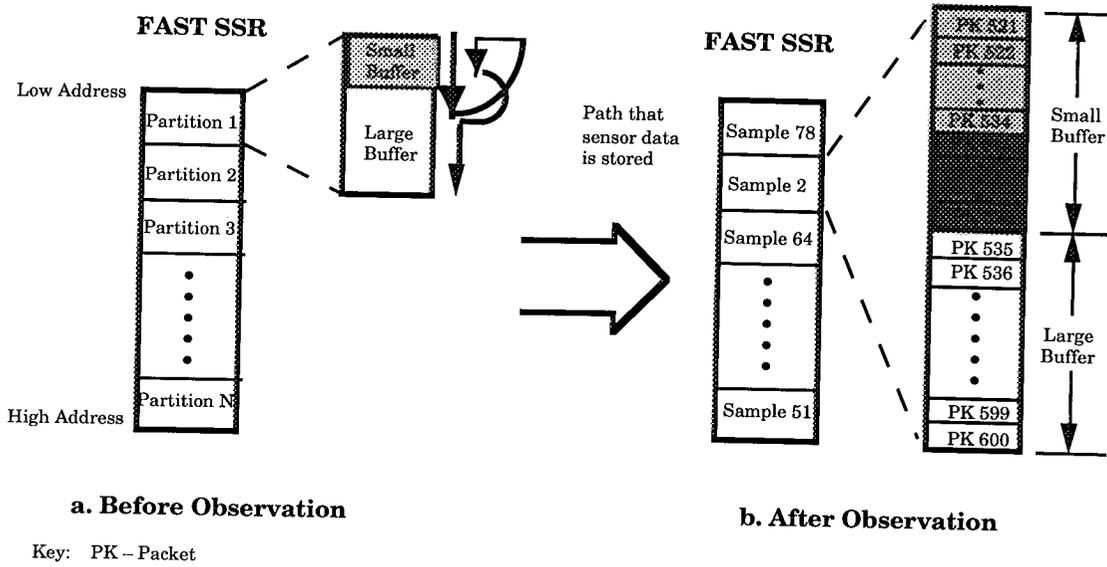
One of the main features of the SSR is that data can be played back in almost any order. This is in sharp contrast with tape recorders that can only be accessed sequentially.

The FAST mission offers an excellent example of how this capability may be implemented. The capacity of the FAST SSR is 128 Mbytes (1 Gigabits). At 8 Megabits per second (Mbps), high-resolution instruments can fill up the SSR in merely 2 minutes! To maximize the use of this limited storage, the FAST Principal Investigator (PI) has developed a complex algorithm to screen sensor data and to take samples, each of which may contain several thousand bytes of sensor data. The algorithm divides the SSR into hundreds of partitions, as shown in Figure 1. Each partition is further divided into a small buffer and a large buffer so that a small portion of data prior to a sensor data sample can be preserved.

When an observation begins, sensor data is taken and compared against a preset threshold. If the data value is less than the threshold, the data is stored sequentially into the small buffer, which operates as a ring buffer. If the data value exceeds the threshold, a sample is taken by jumping to the beginning of the associated large buffer and filling to the end with subsequent sensor data. This process repeats until all available partitions are filled. Then, if samples are still being taken, a comparison is made to all stored samples according to predefined criteria. If the new sample is better, it will overwrite an old sample, which may be anywhere from the first partition to the Nth partition. As the observation continues and the sensor value fluctuates, this scheme will result in fragmented data in the SSR. When the SSR is played back sequentially from a low to high address, the ground system will receive these data fragments that are completely out of time order.

Although ground data systems are designed to handle data fragmentation, the large number of fragments in a data stream will have adverse impact on system performance and resources. As the number of fragments increases, it will take longer for the PPS to merge them together into data sets. More fragments require more system processing and storage resources, and increasing system cost. In extreme cases where millions of data fragments may be generated, ground data systems can be brought nearly to a halt.

There are two remedies to this problem. One is to inform the PI about the impact of data fragmentation and agree to a design limit. In the FAST PPS development, the PI projects 200 data fragments per sensor and agrees to stay within that limit. As a result, there will be about 4000 (200 x 20 science sensors) fragments coming down to the PPS every session. This approach will minimize the impact on flight software development, but limit system's ability to adapt to different data scenarios.



**Figure 1. Simplified FAST SSR Memory Management Example**

An alternative is for the spacecraft data system to maintain a memory map. This map will keep track of time order and the boundaries of the fragments. During a downlink, the spacecraft data system will use information in the memory map to dump recorded data fragments so that their original time order can be preserved. As a result, the ground data processing can be greatly simplified because there will be only one data fragment per sensor instead of thousands. The drawback will be increased complexity in the flight software design.

### 3.2 INTERLEAVE OF REAL-TIME AND PLAYBACK DATA

During ground data processing, separation of real-time and playback data is important for sequence check and time ordering. For missions based on tape recorders, there is a clear distinction between real-time and playback data because real-time data is forward and playback data is reversed. They will never be mixed on a frame-by-frame basis since frame synchronization requires a consistent data direction.

Because playback data from the SSR is also in forward order, it is possible to interleave real-time and playback data into the same data stream. This is desirable by flight operations teams because it allows them to continuously monitor critical spacecraft parameters in real-time while receiving recorded payload data. The problem is that packet sorting can not be performed based only on Path Identifier (ID), a combination of Spacecraft ID (SCID) and Application Process ID (APID), because both real-time and playback data will have the same Path ID. To address this problem, the FAST spacecraft assigns different Virtual Channels (VC) to real-time and playback data. In this manner, the PPS is able to separate real-time and playback data by using the VCID in addition to Path ID, as illustrated in an example in Figure 2. An alternative scheme is to use the REPLAY flag defined by CCSDS in a frame primary header so that the sorting can be accomplished by using the REPLAY flag in addition to Path ID.

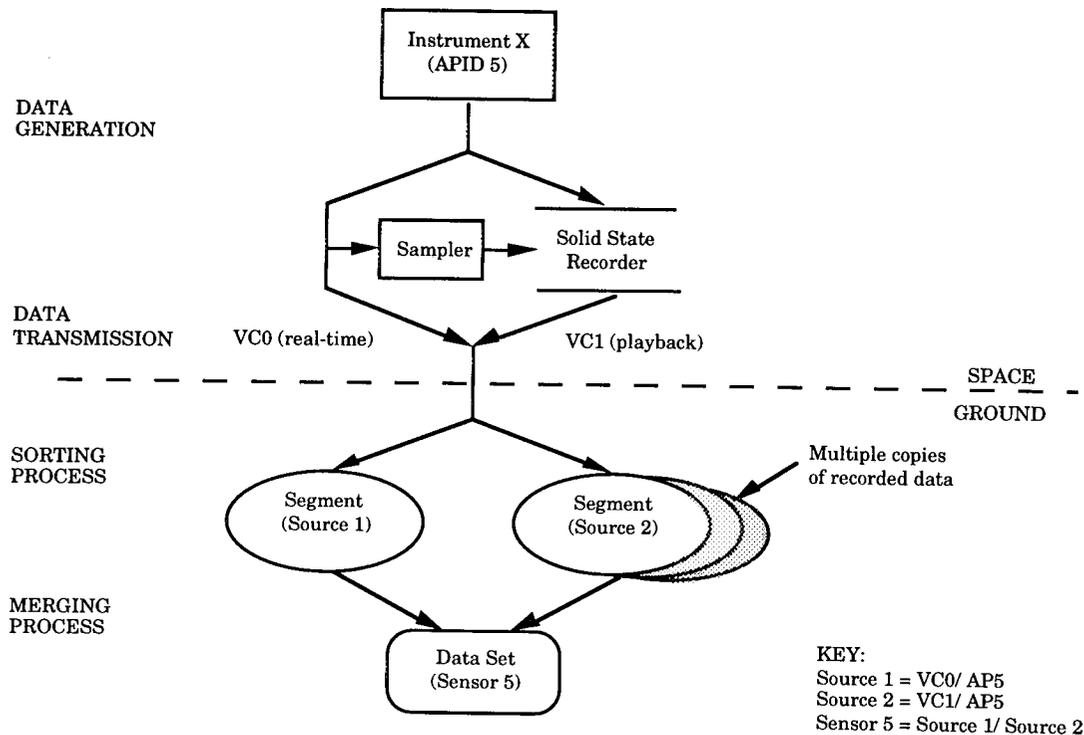


Figure 2. FAST SSR Data Scenario Example

### 3.3 MULTIPLE PLAYBACK OF DATA

As a contingency operation, many missions have planned to redump a certain section of recorded data when the data quality of the first dump is unsatisfactory. Because of the SSRs fast response time, it is feasible for ground controllers to send commands up to require another dump of the same data, or a subset of it based on real-time processing status. As illustrated in Figure 2, multiple copies of the same data will be received by ground data systems during one pass, which increases processing load since all redundant data has to be identified and deleted based on data quality. However, this normally represents a minor impact as only limited attempts can be made for retransmission.

### 3.4 RECORDING SORTED DATA

Another common use of the SSRs flexibility is to sort sensor data as it is stored in the SSR. A typical implementation is to partition the SSR into a number of buffers, one for each sensor or each VC. Consequently, data will be stored in appropriate buffers according to APID or VCID. During a downlink pass, data will be dumped one buffer at a time, i.e., one sensor or VC at a time. There are several advantages to this scheme. First, it may simplify the onboard memory management plan by dividing a large buffer into smaller, but dedicated, buffers. Second, it allows prioritization of data playback. Third, it reduces the processing load of ground systems by performing sensor or VC demultiplexing and sorting functions onboard the spacecraft.

However, the sorting and buffering can change packet rates significantly, and its potential impact on ground systems must not be overlooked. The CCSDS telemetry is a packetized data stream in which many packets, large or small, are multiplexed together. The packet processing load is

proportional to the number of packets per second. Each ground system has a limited ability to process CCSDS source packets as measured in packets per second. Typically, high-rate sensors (e.g. science instruments) generate large packets, and low-rate sensors (e.g. housekeeping sensors) generate small packets. For a fixed data rate, as average packet size decreases, the number of packets per second increases.

For example, assume a fictitious spacecraft has just two sensors, an engineering sensor and an instrument sensor. Every second, the engineering sensor generates a packet of 20 bytes, and the instrument sensor 1000 packets of 1000 bytes. For an 8-Mbps downlink without buffering, this scenario represents a packet rate of 1001 packets per second. On the other hand, if all packets are sorted and buffered, the same 8-Mbps downlink will produce 1000 packets per second for the instrument buffer, and 50,000 packets per second for the engineering buffer! In this scenario, the burst packet rate is increased 50 times. Suppose a ground system is designed to process data in real time at 5000 packets per second – it will still overflow by the later scenario even though it can handle the average packet rate by a wide margin.

This problem was first experienced in the SAMPEX mission. As a result, a recommendation was made to and adopted by the FAST spacecraft development team to interleave recorded engineering data with science data during high-rate downlink with a ratio of 1:5. This represents a small change in flight software, but saves significantly in ground system design and implementation. This is especially true when real-time processing of engineering data is desired.

A second approach calls for packet rate buffering to be performed by the ground data systems. This is very difficult to implement for a number of reasons. It requires expensive resources to buffer frames before packet processing. Packet delivery latency will be much longer and unpredictable. Scheduling of buffered data playback will be very tricky, if not impossible, if an attempt is made to match data packet rates to system capability.

### **3.5 CREATION OF GAPS**

During a ground contact (pass), FAST spacecraft engineering packets are generated at a higher rate and transmitted to ground in real time. As a backup, these packets are also stored in an engineering buffer, in case retransmission is required. However, due to buffer limitation, the packets are sampled and only a subset of them, e.g., one out of four, is stored, causing discontinuity in packet sequence count (e.g., 1, 5, 9, etc.). There are many complications when these packets with large numbers of gaps are received by ground data systems. First: how does the system know this discontinuity is intentionally created, and not caused by transmission errors? Second: how are these packets merged into the ones with sequence counts of different discontinuities? Worse yet: what if packets are sampled at different rates from time to time?

The FAST PPS implemented limited capabilities to handle these cases. Nevertheless, it requires a lot of effort and resources, and still only covers a fraction of the possible scenarios. In general, it is dangerous to tinker with the packet sequence counter, and every effort should be made to avoid this practice if level zero processing is required.

### **3.6 CREATION OF FORMAT ERRORS**

The above discussions treated data scenarios that use the flexibility inherent in the SSR. Ground processing complications will also be encountered if the implementation of the SSR does not maintain any continuity inherent in the format of the data generated. In the case of CCSDS recommended formats, storage and addressing of data in the SSR should support playback while

maintaining frame and byte boundaries. Violation of this condition will result in loss of data, as demonstrated the following example.

In this SSR design, the onboard telemetry generation is running at the very low rate of 6944 bits per second. But even at this low rate, the SSR still can not record and play back at the same time. Like tape recorders, two SSR units of 80 Mbytes each have to be configured with a utilization factor under 50 percent.

Like tape recorders, this SSR is addressable only in 1-Kbyte blocks, rather than on any byte boundaries. Worse, this 1-Kbyte block is asynchronous to byte boundaries, i.e., it may start and stop in the middle of a byte. As a result, each time a write operation begins, a transfer frame is lost and a portion of stored data is corrupted. In addition, this adds significant complexity to ground data processing because many frames may be corrupted with gaps and errors even though communication link quality is good.

#### **4. SUMMARY**

The application of onboard Solid-State Recorders has been discussed from a ground data processing perspective based on experience gained through the SAMPEX, FAST, SWAS, and other space missions. Characteristics of the SSR have been described in contrast with conventional tape recorders. Their advantages and potential negative impacts are detailed using several examples. Many of the data scenarios described have been simulated and tested using an advanced Spacecraft Data Simulator at the DSTD during the development of the FAST PPS [3].

In general, the SSR, with its speed, reliability, and flexibility, offers a technically superior choice for space data systems storage and data processing. With sophisticated management algorithms, the SSR can achieve many objectives that are simply impossible to achieve with conventional tape recorders. However, since the SSR may impose significant performance, resource, and cost impacts on ground data processing systems, the SSR management algorithms should be developed in a coordinated effort with ground system developers. Important parameters such as average and burst packet rates, and maximum number of data fragments per pass should be defined in system specifications to guide both space and ground data system designs. With such an integrated effort, adverse impacts can be avoided or minimized so that the most cost-effective space-ground data systems can be achieved.

#### **5. REFERENCES**

1. Shi, J., Horner, W., Grebowsky, G., Chesney, J., "Fast Auroral Snapshot Explorer (FAST) Packet Processing System (PPS)," Proceedings of International Telemetering Conference, 1993, pp. 445-459.
2. "Packet Telemetry," CCSDS 102.0-B-2, Blue Book, Consultative Committee for Space Data Systems, November, 1992.
3. Shi, J., Gordon, J., Mirchandani, C., Nguyen, D., "Spacecraft Data Simulator for the Test of Level Zero Processing Systems," Space Ops 1994.

## 6. NOMENCLATURE

ACE	Advanced Composition Explorer
APID	Application Process Identifier
CCSDS	Consultative Committee for Space Data Systems
DSTD	Data Systems Technology Division
EOS-AM	Earth Observing System AM
FAST	Fast Auroral Snapshot Explorer
GSFC	Goddard Space Flight Center
LZP	Level Zero Processing
PI	Principal Investigator
RAM	Random Access Memory
SAMPEX	Solar Anomalous and Magnetospheric Particle Explorer
SCID	Spacecraft Identifier
SMEX	Small Explorer
SWAS	Submillimeter Wave Astronomy Satellite
SSR	Solid-State Recorder
VC	Virtual Channel
VCID	Virtual Channel Identifier